

6. (Amended-Clean Text) A computing environment as claimed in claim 1 wherein said evolutionary operations include the selective deletion of objects from within said process.

7. (Amended-Clean Text) A computing environment as claimed in claim 1 wherein said evolutionary operations include the selective loading or reloading of objects into said process.

12. (Amended-Clean Text) A computing environment as claimed in claim 9 wherein said construct is provided with an authorising signature.

13. (Amended-Clean Text) A computing environment as claimed in claim 9 wherein after said construct is transferred the second process stored within said construct is caused to be activated within said first process.

14. (Amended-Clean Text) A computing environment as claimed in claim 9 wherein after said construct is transferred the first process is suspended and the second process stored within said construct is be activated, and when the second process is concluded the data and program modules of the second process are added to the first process and the first process is re-activated.

15. (Amended-Clean Text) A computing environment as claimed in claim 9 wherein after said first process terminates at least some of the data and/or program modules from said first process are added to the second process stored in said construct and said second process is then activated.

19. (Amended-Clean Text) A computing environment as claimed in claim 16 wherein said data and said program modules from said second process are copied into said first process.

20. (Amended-Clean Text) A computing environment as claimed in claim 8 wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said first process will override the data and/or program modules of said second process.

21. (Amended-Clean Text) A computing environment as claimed in claim 8 wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said second process will override the data and/or program modules of said first process.

P17529.A01

25. (Amended-Clean Text) A computing environment as claimed in claim 22 wherein a said process is subject to an evolutionary operation that allows the process to run in the second hardware component.

26. (Amended-Clean Text) A computing environment as claimed in claim 22 wherein said second hardware component is a memory storage device.

31. (Amended-Clean Text) A method as claimed in claim 28 wherein said construct is provided with an authorising signature.

32. (Amended-Clean Text) A method as claimed in claim 27 wherein a process is modified by the selective deletion of objects from within the process.

33. (Amended-Clean Text) A method as claimed in claim 27 wherein a process is modified by the selective loading or reloading of objects into a said process.

38. (Amended-Clean Text) A method as claimed in claim 35 wherein said construct is formed with an authorising signature.

39. (Amended-Clean Text) A method as claimed in claim 35 wherein after said construct is transferred the second process stored within said construct is caused to be activated within said first process.

40. (Amended-Clean Text) A method as claimed in claim 35 wherein after said construct is transferred the first process is suspended and the second process stored within said construct is activated, and when the second process is concluded the data and program modules of the second process are added to the first process and the first process is re-activated.

41. (Amended-Clean Text) A method as claimed in claim 35 wherein after said first process terminates at least some of the data and/or program modules from said first process are added to the second process stored in said construct and said second process is then activated.

45. (Amended-Clean Text) A method as claimed in claim 42 wherein said at least some data and/or program modules from said second process are copied into said first process.